

NOTE SULLO SVOLGIMENTO DELLA PROVA:

- Fare partire il Dev-C++ (dal Menù **Avvio** (o **Start**) nella barra degli strumenti in fondo allo schermo, selezionare Programmi e quindi Dev-C++);
- attraverso il menu **File->Open...** (o **apri** se il Dev-C++ è in italiano) aprire il progetto *APRIQUESTO.dev* presente nel directory *c:\esame\esaInf*;
- scrivere le funzioni richieste nel file *compito.cpp* già presente nel progetto;
- salvare spesso in modo da non perdere il lavoro nel caso in cui il PC abbia un malfunzionamento (per salvare premere il bottone a forma di dischetto del Dev-C++);
- per una corretta stampa dell'elaborato bisogna mantenere il codice entro i margini imposti dall'ambiente Dev-C++ (linea verticale presente alla destra della pagina).

Ufficio postale

Un programma per la gestione degli sportelli di un ufficio postale può essere realizzato usando le seguenti costanti e i seguenti tipi:

```
const int MAXS = 100;
const int MAXC = 20;
const int MAXN = 10;
const float DUR_POSTA = 30.0f;
const float DUR_FIN = 45.0f;
```

```
enum Oper {POSTA, FINANZA};
```

```
struct Cliente {
    char nome[MAXS];
    Oper oper;
};
```

```
struct Sportello {
    char id;
    int back, front;
    Cliente cli[MAXC];
};
```

```
struct UfficioPostale {
    Sportello sport[MAXN];
    int numSport;
};
```

Il tipo `Cliente` contiene le informazioni relative a un cliente: il suo nome e il tipo di operazione che deve fare. Le operazioni possibili sono due, finanza e posta, e sono specificate dal tipo enumerato `Oper`. Ogni sportello mantiene una coda di clienti ed è rappresentato dall'omonima struttura. I campi `front` e `back` indicano le posizioni dell'array `cli` in cui eseguire le prossime operazioni di estrazione e inserimento; il campo `id` contiene un carattere che identifica in modo univoco il singolo sportello (con il solo vincolo che un identificatore di sportello non può essere uguale al carattere 'X'). Un ufficio postale contiene tanti sportelli (al più `MAXN`) e li memorizza nell'array `sport`. Il campo `numSport` indica il numero effettivo di sportelli di quell'ufficio postale.

Scrivere il corpo delle seguenti funzioni C++.

1. **char assegna(UfficioPostale* pu, Cliente c)** inserisce il cliente *c* nella coda dello sportello in cui sono presenti, al momento, meno clienti. Se ci sono più sportelli con lo stesso numero minimo di clienti se ne può scegliere uno qualunque. La funzione restituisce l'identificatore dello sportello in cui il cliente è stato accodato. Viene restituito il valore 'X' nel caso in cui tutte le code di tutti gli sportelli siano piene e il cliente non possa essere inserito.
2. **bool chiusuraSportello(UfficioPostale* pu, char c)** chiude lo sportello avente l'id specificato dal parametro *c*. Tutti i clienti dello sportello che viene chiuso devono essere riassegnati ad altri sportelli (finendo in fondo alle code dei nuovi sportelli). La funzione restituisce *true* nel caso in cui lo sportello da chiudere sia effettivamente presente nell'ufficio postale e che tutti i clienti possano essere riassegnati ad altri sportelli. La funzione restituisce *false* nel caso in cui non esista alcuno sportello con id pari a *c* o nel caso in cui non tutti i clienti dello sportello appena chiuso possano essere riassegnati (tali clienti spariranno, di fatto, dal sistema). Lo sportello chiuso deve essere anche rimosso dall'ufficio postale.
3. **void stampaStato(const UfficioPostale* pu)** stampa lo stato dell'ufficio postale. Ogni sportello viene stampato su una riga contenente l'id dello sportello e a seguire tante lettere quanti sono i clienti in coda a tale sportello. Ogni cliente viene visualizzato come una lettera, *p* o *f*, a seconda che debba eseguire una operazione di tipo POSTA o FINANZA. Esempio:

F fffp
B ppp
G pp
E pff
4. **bool desiste(UfficioPostale* pu, const char n[])** il cliente avente nome *n* desiste e se ne va dall'ufficio postale. Si supponga che non ci siano casi di omonimia. La coda dello sportello in cui era accodato il cliente che desiste deve essere ricompattata in modo opportuno. La funzione restituisce *true* se un cliente con tale nome esiste e viene rimosso dal sistema, *false* altrimenti.
5. **bool codaMax(const UfficioPostale* pu, char* pid, float* pdur)** trova lo sportello avente la coda di operazioni più lunga da smaltire. "Più lunga" deve essere inteso nel senso di durata temporale supponendo che le singole operazioni di tipo POSTA e FINANZA abbiano la durata specificata dalle costanti DUR_POSTA e DUR_FIN. La funzione restituisce al chiamante l'id dello sportello con coda di durata più lunga riempiendo l'oggetto puntato da *pid*; la durata viene restituita riempiendo l'oggetto puntato da *pdur*. La funzione restituisce *true* nel caso in cui l'operazione possa essere eseguita (c'è almeno uno sportello), *false* altrimenti.